REMARKS

Claims 1-52 are pending in the application.

Claims 1-14, 16-29 and 31-52 were rejected under 35 U.S.C. Section 102(e) as being anticipated by Teranishi et al. (USPN 6,117,183) (hereinafter referred to simply as "Teranishi"). Furthermore, Claims 15 and 30 were rejected under 35 U.S.C. Section 103(a) as being unpatentable over Teranishi in view of McKaskle et al. (USPN 5,481,741) (hereinafter referred to simply as "McKaskle"). Applicant respectfully traverses these rejections based on the following reasoning.

Each of the independent claims recites something similar to:

> "displaying a first hardware device node in a graphical program in response to user input, wherein the graphical program comprises a plurality of interconnected nodes or icons, wherein the plurality of interconnected nodes or icons visually indicate functionality of the graphical program". (*Emphasis added*)

Applicant's specification defines the term "graphical program" in part at page 3, lines 17-27:

> In response to the user constructing a diagram or graphical program using the block diagram editor, data structures may be automatically constructed which characterize an execution procedure which corresponds to the displayed procedure. The graphical program may be compiled or interpreted by a computer. (*Emphasis added*)
>
> Therefore, Kodosky et al teaches a graphical programming environment wherein a user places or manipulates icons and interconnects or "wires up" the icons in a block diagram using a block diagram editor to create a graphical "program." A graphical program for measuring, controlling, or modeling devices, such as instruments, processes or industrial automation hardware, or for modeling or simulating devices, may be referred to as a virtual instrument (VI). Thus, a user can create a computer program solely by using a graphically based programming environment. (*Emphasis added*)

Thus, it is apparent that Teranishi never teaches or suggests anything close to a graphical program or to a node in a graphical program.

The Examiner may be tempted to identify the component placement diagram of Figure 2 with a graphical program as recited in the claims. However, such an identification would be mistaken because Teranishi never suggests that the displayed items L0 through L12 and their interconnections visually indicate the functionality of a graphical program. Furthermore, there is no sense in which the component placement diagram characterizes a program which can be compiled or interpreted by a computer.

The Examiner states that "Teranishi teaches a graphical program because Teranishi discloses a graphical user interface that allows a user to move, update, and associate virtual objects. (column 4, lines 51-68)". Thus, the Examiner appears to be assuming that any program (e.g., a program such as a graphical user interface) that is configured to manipulate graphical objects on a display is a graphical program. This assumption is not consistent with Applicant's definition of graphical program as recited in the claims and as taught in the specification. A graphical program is a program whose functionality is visually indicated by a plurality of interconnected nodes or icons as recited in the claims. Teranishi never suggests a graphical user interface whose functionality is visually indicated by a plurality of interconnected nodes or icons.

Therefore, the independent claims and their dependents are patentably distinguished over the cited references at least for the reasons given above.

Furthermore, Teranishi never suggests *propagating information from a first hardware device node to a second hardware device node, where the information specifies the hardware device with which the first hardware device node is associated* as recited in Claims 1, 18 and 32. Regarding this feature, the Examiner states:

> "Teranishi teaches this limitation because Teranishi locates the problematic sign[*al*] path within a circuit. (column 4, lines 1-25) In order the [*sic*] locate the problematic path, Teranishi simulates the circuit (column 6, lines 17-30) and its components and that requires propagating information from one hardware device node to another device. (column 6, lines 46-58, column 8, lines 12-53)"

Applicant respectfully disagrees. Teranishi teaches extracting (i.e., identifying) a signal path that does not satisfy a target cycle time as an error path based on signal path data (SPD) 44 containing a delay value calculated for each signal path from the component placement data (Col. 4, lines 1-5). This extraction process does not in any way suggest

the existence or use of nodes in a graphical program. Thus, it is not possible that the extraction process could suggest *propagating information from a first hardware device node to a second hardware device node, where the information specifies the hardware device with which the first hardware device node is associated* as recited in Claims 1, 18 and 32.

The Examiner states that "Teranishi simulates the circuit". Applicant notes that Teranishi never suggests simulating the functional behavior of the displayed elements (such as elements L0 through L12 of Figure 2). Teranishi appears to be concerned primarily with the computation and display of time delays associated with worst case paths in order to enable the user to move components in a way that decreases the worst case time delays (See Col. 4, lines 6-58 and especially Figure 3).

Therefore, Claims 1, 18 and 32, and their dependents, are additionally distinguished over the cited references.

Moreover, Teranishi never suggests any of the following features recited in Claims 10, 26 and 39:

> "associating the first hardware device node with a first hardware device class in response to user input";
>
> "selecting a method or property of the first hardware device class for the first hardware device node in response to user input";
>
> "changing the first hardware device node to have an association with a second hardware device class in response to user input"; and
>
> "performing type checking to determine whether the method or property is valid for the second hardware device class, in response to said changing the first hardware device node to have an association with the second hardware device class".

The notion of hardware device class is entirely missing from Teranishi. Thus, it is not possible to reasonably construe Teranishi as teaching the operation of associating a hardware device node with a hardware device class, or, the operation of selecting a method or property of a hardware device class as recited in Claims 10, 26 and 39. Therefore, Claims 10, 26 and 39, and their dependents are additionally distinguished over the cited references.

CONCLUSION

In light of the foregoing amendments and remarks, Applicant submits the application is now in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 50-1505/5150-52100/JCH.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

Respectfully submitted,

Mark K. Brightwell
Reg. No. 47,446
AGENT FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: July 7, 2006 JCH/MKB